**find the safest road**

**Time Limit: 10000/5000 MS (Java/Others)    Memory Limit: 32768/32768 K (Java/Others)  
Total Submission(s): 10675    Accepted Submission(s): 3778**

Problem Description

XX 星球有很多城市，每个城市之间有一条或多条飞行通道，但是并不是所有的路都是很安全的，每一条路有一个安全系数s,s是在 0 和 1 间的实数(包括0，1)，一条从u 到 v 的通道P 的安全度为Safe(P) = s(e1)\*s(e2)…\*s(ek) e1,e2,ek是P 上的边 ，现在8600 想出去旅游，面对这这么多的路，他想找一条最安全的路。但是8600 的数学不好，想请你帮忙 ^\_^

Input

输入包括多个测试实例，每个实例包括：  
第一行：n。n表示城市的个数n<=1000;  
接着是一个n\*n的矩阵表示两个城市之间的安全系数，(0可以理解为那两个城市之间没有直接的通道)  
接着是Q个8600要旅游的路线,每行有两个数字，表示8600所在的城市和要去的城市

Output

如果86无法达到他的目的地，输出"What a pity!",  
其他的输出这两个城市之间的最安全道路的安全系数,保留三位小数。

Sample Input

3

1 0.5 0.5

0.5 1 0.4

0.5 0.4 1

3

1 2

2 3

1 3

Sample Output

0.500

0.400

0.500

Author

ailyanlu

分析：Dijksra的变形 把最短路换成最大概率就可以了 要涉及到小数 用double

但是 还要注意的是 中间的变量 max也要定义成 double 这个地方 刚开始 坑了好几次

AC代码：

#include <iostream>

#include <cstdio>

#include <cstring>

#define MAX 10001

#define INF 0x3f3f3f3f

using namespace std;

int n,m;

double map[MAX][MAX];

void dijkstra(int a,int b)

{

double dist[MAX],s[MAX];

int u = a;

double max;

memset(s,0,sizeof(s));

for(int i = 1; i <= n; i++)

{

dist[i] = map[a][i];

s[i] = 0;

}

s[a] = 1;

for(int i = 1; i <= n; i++)

{

max = 0;

for(int j = 1; j <= n; j++)

{

if(s[j]==0&&dist[j]>max)

{

max = dist[j];

u = j;

}

}

s[u] = 1;

for(int j = 1; j <= n; j++)

{

if(s[j]==0)

{

if(map[u][j]>0&&dist[u]\*map[u][j]>dist[j])

{

dist[j] = dist[u] \* map[u][j];

}

}

}

}

if(dist[b] == 0)

printf("What a pity!\n");

else

printf("%0.3lf\n",dist[b]);

}

int main()

{

while(~scanf("%d",&n)&&n)

{

for(int i = 1; i <= n; i++)

{

for(int j = 1; j <= n; j++)

{

scanf("%lf",&map[i][j]);

}

}

int a,b;

scanf("%d",&m);

for(int i = 1; i <= m; i++)

{

scanf("%d%d",&a,&b);

dijkstra(a,b);

}

}

return 0;

}